

PREDIKSI PENGGUNAAN *BANDWIDTH* MENGGUNAKAN *ELMAN RECURRENT NEURAL NETWORK*

Jefri Radjabaycolle¹, Reza Pulungan²

^{1,2}Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
e-mail: ¹jefriradja@gmail.com, ²pulungan@ugm.ac.id

Abstrak

Jaringan Syaraf Tiruan (JST) sering dipakai dalam menyelesaikan permasalahan tertentu seperti prediksi, klasifikasi, dan pengolahan data. Berdasarkan hal tersebut, dalam penelitian ini mencoba menerapkan JST untuk menangani permasalahan dalam prediksi penggunaan *bandwidth*. Sistem yang dikembangkan dapat digunakan untuk memprediksi penggunaan *bandwidth* dengan menerapkan *Elman Recurrent Neural Network* (ERNN). Struktur Elman dipilih karena dapat membuat iterasi jauh lebih cepat sehingga memudahkan proses konvergensi. Vektor input yang digunakan menggunakan *windows size*. Hasil penelitian dengan menggunakan target *error* sebesar 0.001 menunjukkan nilai MSE terkecil yaitu pada *windows size* 11 dengan nilai 0.002833. Kemudian dengan menggunakan 13 *neuron* pada *hidden layer* diperoleh nilai *error* paling optimal (*minimum error*) sebesar 0.003725.

Kata kunci : Prediksi, *Bandwidth*, jaringan syaraf tiruan (JST), *elman recurrent neural network*

PREDICTIONS OF BANDWIDTH USING ELMAN RECURRENT NEURAL NETWORK

Abstract

Artificial Neural Network (ANN) can be used to solve specific problems such as prediction, classification, and data processing. Accordingly, in this study tries to apply neural networks to deal with the problem of prediction bandwidth usage. The system developed can be used to predict the bandwidth usage by applying Elman Recurrent Neural Network (ERNN). Elman structure chosen because it can make much faster iterations so as to facilitate the process of convergence. Input vector used to use *windows size*. The results using a target error of 0001 showed the smallest MSE value is on the *windows size* 11 with a value of 0.002833. Then, using 13 neurons in the hidden Layer error values obtained most optimal (*minimum error*) amounted to 0.003725.

Keywords : Prediction, Bandwidth, artificial neural network (ANN), *elman recurrent neural network*

1. Pendahuluan

Bandwidth merupakan ukuran jumlah data yang dapat melakukan perjalanan lebih dari satu sistem komunikasi yang dialokasikan dalam rentang waktu atau disebut juga sebagai kecepatan data. *Bandwidth* juga memiliki arti bahwa semakin besar *bandwidth* bisa menghasilkan komunikasi yang lebih cepat. Ketersediaan *bandwidth* jaringan merupakan faktor penting dalam memilih layanan web. Pada dasarnya besarnya kebutuhan *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kebutuhan *bandwidth*, umumnya akan diikuti oleh kinerja yang lebih baik. Salah satu solusi yang paling efektif untuk mengatasinya adalah dengan mengelola pemakaian *bandwidth* yang menghasilkan suatu kualitas layanan lalu lintas aliran data yang baik dan berkualitas.

Kemampuan untuk mengantisipasi kebutuhan *bandwidth* sangat penting untuk layanan yang efisien dan pengambilan keputusan cerdas dalam menghadapi perkembangan lalu lintas yang cepat dan perubahan pola lalu lintas [1]. Pada dasarnya besarnya kebutuhan *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kebutuhan *bandwidth*, umumnya akan diikuti oleh kinerja yang lebih baik, meskipun kinerja keseluruhan juga tergantung pada faktor-faktor lain, misalnya *latency* yaitu waktu tunda antara masa sebuah

perangkat meminta akses ke jaringan dan masa perangkat itu memberi izin untuk melakukan transmisi .

Khususnya yang terjadi di lingkungan kampus Universitas Pattimura, terdapat beberapa keluhan dari pengguna jaringan internet. Mahasiswa maupun karyawan dan dosen di Universitas Pattimura mengeluhkan adanya koneksi internet yang dirasa kurang sesuai dengan apa yang diharapkan, misalnya dari segi kecepatan ataupun kestabilan koneksi. Kecepatan koneksi yang didapatkan oleh pengguna tidak hanya dipengaruhi oleh alokasi *bandwidth* yang diberikan oleh pihak kampus tetapi dipengaruhi juga oleh alokasi *bandwidth* yang disediakan oleh penyedia layanan *server* yang diakses atau dituju oleh pengguna. Dari pihak administrator jaringan di Universitas Pattimura, diperoleh informasi total *bandwidth* yang dimiliki pihak kampus dengan pembagian alokasi *bandwidth* berdasarkan jumlah pengguna pada setiap departemen atau bagian yang ada. Secara garis besar ada beberapa faktor yang harus dipertimbangkan dalam menentukan alokasi *bandwidth* yaitu jumlah departemen beserta karyawan yang ada dalam setiap departemen, jumlah dosen, jumlah mahasiswa, serta alokasi khusus yang diperlukan untuk layanan *server* untuk mendukung kegiatan belajar mengajar di kampus. Permasalahan yang muncul dari pembagian *bandwidth* yang sudah dilaksanakan adalah sudah tepatkah pembagian *bandwidth* yang diimplementasikan untuk saat ini, perlu dilakukan perubahan atau tidak untuk pembagian *bandwidth* nya sehingga biaya yang dikeluarkan untuk sewa *bandwidth* dapat menjadi efektif dan efisien.

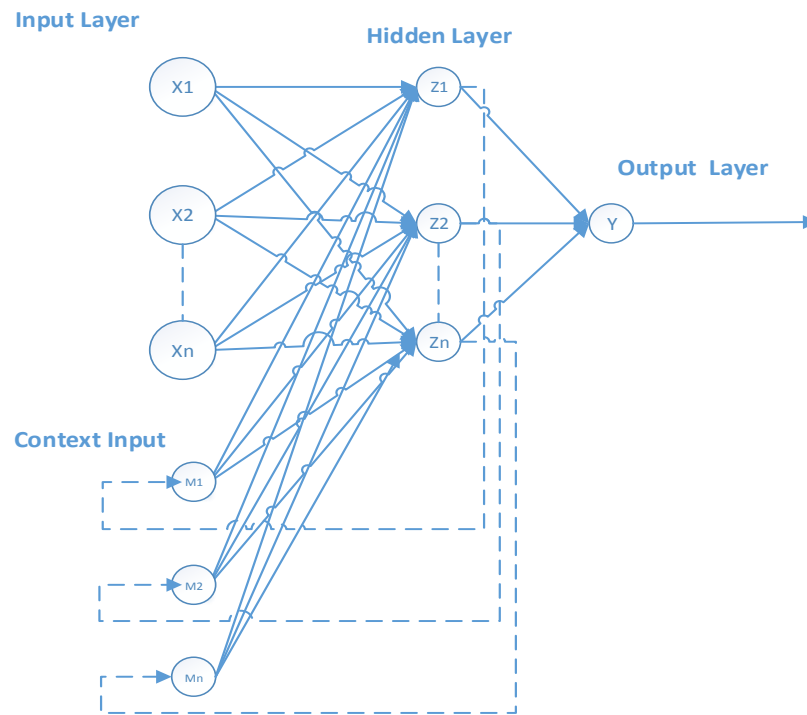
Berangkat dari permasalahan yang sudah disebutkan diatas, diperlukan adanya analisa lebih mendalam mengenai berapa kebutuhan *bandwidth* yang harus disediakan oleh pihak kampus sehingga layanan koneksi internet maupun *Local Area Network* (LAN) dapat berjalan dan mendukung semua aktifitas yang memerlukan kondisi *network* dengan layanan yang sesuai. Untuk membantu pihak manajemen menentukan alokasi yang tepat untuk sewa *bandwidth* maka dibutuhkan suatu sistem yang dapat memprediksi penggunaan *bandwidth* menggunakan jaringan syaraf tiruan dengan algoritma *Elman Recurrent Neural Network*. Struktur Elman dipilih karena dapat membuat iterasi jauh lebih cepat dan konvergensi akan menjadi lebih cepat yang disebabkan oleh jaringan *Elman* memiliki *feedback* yang menawarkan proses pembelajaran yang lebih cepat pada jaringan [2]. Kemampuan belajar yang dimilikinya dapat dilatih untuk mempelajari dan menganalisa pola data masa lalu dan berusaha mencari suatu formula atau fungsi yang akan menghubungkan pola data masa lalu dengan keluaran yang diinginkan pada saat ini [3].

2. Metode Penelitian

2.1 Arsitektur ERRN (*Elman Recurrent Neural Network*)

Recurrent Neural Network adalah salah satu bagian dari model *artificial neural network* yang mempunyai *feedback* dari keluaran *hidden Layer* ke masukan *input Layer* . *Recurrent Neural Network* mempunyai struktur dan algoritma pelatihan yang lebih kompleks dibandingkan *feed-forward neural network*. Pada *Recurrent Neural Network*, *output* dari *network* digunakan kembali sebagai *input network*. *Elman Recurrent Neural* disebut *partial recurrent neural network* karena *recurrent weight* adalah tetap [4]. *Network* ditambahkan di *layer context* sebagai tambahan *layer* proses. Biasanya *network* ini merupakan *feedforward neural network*. Pelatihan dilakukan pada hubungan *forward*. Sedangkan hubungan *backward* dari *output Layer* ke *input Layer* tidak dapat dilakukan pelatihan. *Layer context* digunakan untuk mengingat status terakhir dari *hidden layer* . *Output* dari *network* tergantung dari status sebelum maupun status *network* pada saat ini. Kemampuan dalam mengingat status terakhir menjadikan *network* ini memiliki memori yang dinamis. Arsitektur *Elman Neural Network* hampir sama dengan arsitektur *feedforward backpropagation*, namun ditambah dengan *layer context* untuk menampung hasil *output* dari *hidden layer* . *Layer* akan di *update* tidak hanya pada *input* jaringan saja tapi juga dengan aktivasi *forward propagation* sebelumnya.

Elman Recurrent Neural Network merupakan variasi dari *Multi Layer Perceptron*. Akan tetapi pada *Elman Recurrent Neural Network* terdapat beberapa *node* yang posisinya berdekatan dengan *input layer* yang berhubungan dengan *hidden layer*. *Node-node* tersebut mengandung isi dari salah satu *layer* yang telah dilatih sebelumnya. Pada prinsipnya, masukan disebarkan secara *feed forward* yang kemudian diberikan suatu *learning rule*. Jaringan jenis dapat memelihara suatu urutan keadaan dan mengijikannya untuk melakukan beberapa pekerjaan sekaligus, contohnya seperti *sequence prediction* yang berada diluar kemampuan *Multi Layer Perceptron* [4]. Masukan tidak hanya nilai dari luar jaringan, tetapi ditambah dengan nilai keluaran dari *neuron* tersembunyi dari propagasi sebelumnya. Gambar 1 merupakan struktur *Elman Recurrent Network*.

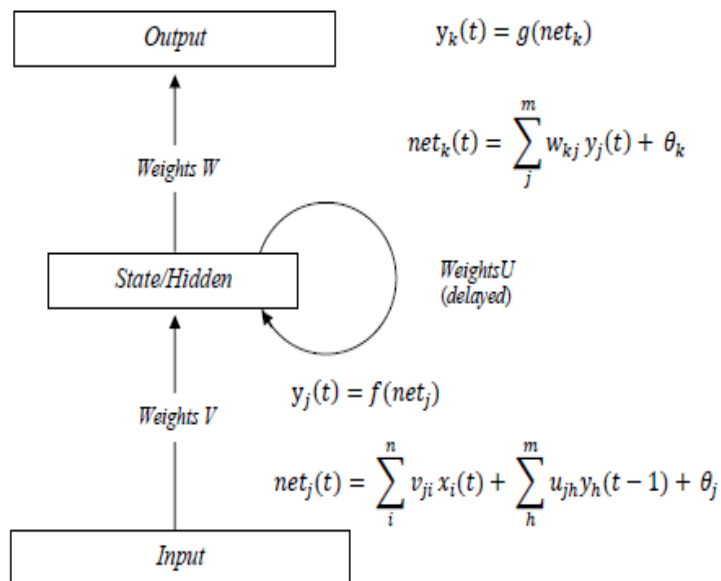


Gambar 1. Arsitektur Elman Recurrent Neural Network

Model jaringan syaraf tiruan untuk prediksi pemakaian dan penggunaan *bandwidth* menggunakan struktur jaringan syaraf tiruan 3 layer /lapisan. Struktur jaringan syaraf tiruan 3 layer terdiri dari sebuah *layer input*, sebuah *layer hidden*, sebuah *layer output* dan *context input*.

2.2 Algoritma Elman Recurrent Neural Network

Jaringan *Elman vector input* akan dipropagasikan melalui *layer bobot* dan akan dikombinasikan dengan aktifasi sebelumnya memlaui *layer bobot recurrent* tambahan yaitu *U* seperti pada Gambar 2 dimana *U* adalah bobot delay.



Gambar 2. Jaringan Elman

Setiap kali sebuah pattern disajikan, unit akan menghitung aktifasinya seperti halnya pada jaringan *feedforward*. Adapun tahapan dalam algoritma *JST Recurrent Elman* adalah sebagai berikut :

Tahap 1 : *Training* pola input atau vector saat time (t) seperti persamaan (1)

$$y_t(t) = f(\text{net}_j) \quad (1)$$

dengan jaringan *recurrent* seperti pada persamaan (2)

$$\text{net}_j(t) = \sum_i^n v_{ji}x_i(t) + \sum_h^m u_{jh}y_h(t-1) + \theta_j \quad (2)$$

Tahap 2 : *Output* jaringan ditentukan oleh lapisan *context* dan satu set *output* berbobot (w) seperti persamaan (3), yang kemudian dapat dijabarkan menjadi persamaan (4).

$$y_k(t) = g(\text{net}_k(t)) \quad (3)$$

$$\text{net}_k(t) = \sum_j^m w_{kj}y_j(t)\theta_k \quad (4)$$

Tahap 3 : *Back propagation* untuk jaringan *recurrent*, Setiap bobot dimodifikasi, dimana fungsi biaya (kesalahan) sehubungan dengan bobot dihitung dan kemudian disesuaikan. Fungsi biaya yang digunakan dalam penelitian ini adalah *Mean Square Error* (MSE) pada persamaan (5)

$$MSE = \frac{1}{n} \sum_p^n \sum_k^m (d_{pk} - y_{pk})^2 \quad (5)$$

keterangan :

d : ouput yang diinginkan

n : jumlah total sampel

m : jumlah *node output*

Tahap 4 : Menurut *gradient descent*, setiap perubahan bobot dalam jaringan harus sebanding dengan *gradient* negatif dari biaya sehubungan dengan tertentu yang tertarik untuk dimodifikasi. η adalah sebuah *learning rate* seperti pada persamaan (6).

$$\Delta w = -\eta \frac{\partial C}{\partial w} \quad (6)$$

Tahap 5 : Menggunakan aturan rantai dari diferensiasi, bobot berkaitan dengan *error* pada *output* dapat dinyatakan dengan persamaan (7).

$$\delta_{pk} = -\frac{\partial C}{\partial y_{pk}} \frac{\partial y_{pk}}{\partial \text{net}_{pk}} = (d_{pk} - y_{pk})g'(y_{pk}) \quad (7)$$

Sedangkan *error* pada *hidden layer* dinyatakan dengan persamaan (8)

$$\delta_{pj} = -\left(\sum_k^m \frac{\partial C}{\partial y_{pk}} \frac{\partial y_{pk}}{\partial \text{net}_{pk}} \frac{\partial \text{net}_{pk}}{\partial y_{pj}} \right) \frac{\partial y_{pj}}{\partial \text{net}_{pj}} = \sum_{k=1}^m \delta_{pk} w_{kj} f'(y_{pj}) \quad (8)$$

Dimana m adalah *neuron* pada lapisan *output*.

Tahap 6 : Perubahan bobot untuk *output* dinyatakan dengan persamaan (9). Sedangkan persamaan (10) untuk perubahan bobot *input*.

$$\Delta w_{kj} = \eta \sum_p^n \delta_{pk} y_{pj} \quad (9)$$

$$\Delta v_{ji} = \eta \sum_p^n \delta_{pj} y_{pi} \quad (10)$$

Tahap 7 : Berdasarkan komponen waktu, perubahan bobot *recurrent* dapat dinyatakan seperti persamaan (11).

$$\Delta u_{jh} = \eta \sum_p^n \delta_{pj}(t) y_{ph}(t-1) \quad (11)$$

Tahap 8 : Pelatihan akan berhenti apabila *error* < target *error*.

2. 3 Teori Prediksi

Secara umum pengertian prediksi adalah tafsiran. Namun dengan menggunakan teknik-teknik tertentu maka prediksi bukan hanya sekedar tafsiran. Ada beberapa definisi tentang prediksi, diantaranya :

1. Prediksi diartikan sebagai penggunaan teknik-teknik statistik dalam bentuk gambaran masa depan berdasarkan pengolahan angka angka historis.
2. Prediksi merupakan bagian integral dari kegiatan pengambilan keputusan manajemen.
3. Prediksi adalah peramalan, rencana, atau estimasi kejadian masa depan yang tidak pasti. Selain itu prediksi juga dapat diartikan sebagai penggunaan teknik-teknik statistik dalam membentuk gambaran masa depan berdasarkan pengolahan angka-angka historis.

Metode prediksi merupakan cara memperkirakan apa yang akan terjadi pada masa depan secara sistematis dan pragmatis atas dasar data yang relevan pada masa yang lalu, sehingga dengan demikian metode prediksi diharapkan dapat memberikan objektivitas yang lebih besar. Selain itu metode prediksi dapat memberikan cara pengerjaan yang teratur dan terarah, dengan demikian dapat dimungkinkannya penggunaan teknik penganalisaan yang lebih maju. Dengan penggunaan teknik-teknik tersebut maka diharapkan dapat memberikan tingkat kepercayaan dan keyakinan yang lebih besar, karena dapat diuji penyimpangan atau deviasi yang terjadi secara ilmiah.

2 3.1 Jenis Prediksi

Berdasarkan sifatnya, prediksi dibedakan atas dua macam yaitu :

- a. Prediksi kualitatif adalah prediksi yang didasarkan atas pendapat suatu pihak, dan datanya tidak bisa direpresentasikan secara tegas menjadi suatu angka atau nilai. Hasil prediksi yang dibuat sangat bergantung pada orang yang menyusunnya. Hal ini penting karena hasil prediksi tersebut ditentukan berdasarkan pemikiran yang instuisi, pendapat dan pengetahuan serta pengalaman penyusunnya.
- b. Prediksi kuantitatif adalah prediksi yang didasarkan atas data kuantitatif masa lalu (data historis) dan dapat dibuat dalam bentuk angka yang biasa disebut sebagai data *time series*

Hasil prediksi yang dibuat sangat bergantung pada metode yang dipergunakan dalam prediksi tersebut. Baik tidaknya metode yang dipergunakan ditentukan oleh perbedaan atau penyimpangan antara hasil ramalan dengan kenyataan yang terjadi. Semakin penyimpangan antara hasil ramalan dengan kenyataan yang akan terjadi maka semakin baik pula metode yang digunakan.

2.3.2 Pengukuran Prediksi

Teknik prediksi tidak selamanya selalu tepat karena teknik prediksi yang digunakan belum tentu sesuai dengan sifat datanya, atau disebabkan oleh kondisi di luar bisnis yang mengharuskan bisnis itu menyesuaikan diri. Oleh karena itu, perlu diadakan pengawasan prediksi sehingga dapat diketahui sesuai atau tidaknya teknik prediksi yang digunakan. Sehingga dapat dipilih dan ditentukan teknik prediksi yang lebih sesuai dengan cara menentukan batas toleransi prediksi atas penyimpangan yang terjadi.

Pada prinsipnya, pengawasan prediksi dilakukan dengan membandingkan hasil prediksi dengan kenyataan yang terjadi. Penggunaan teknik prediksi yang menghasilkan penyimpangan terkecil adalah teknik prediksi yang paling sesuai untuk digunakan.

Besarnya *error* prediksi dihitung dengan mengurangi data riil dengan besarnya ramalan pada persamaan (12).

$$Error(E) = X_t - F_t \quad (12)$$

Keterangan

X_t = Data riil periode ke- t

F_t = Ramalan Periode ke- t

2. 4 Proses Jaringan Syaraf Tiruan

Penerapan jaringan syaraf tiruan dalam permasalahan prediksi penggunaan *bandwidth* meliputi penentuan komponen-komponen jaringan syaraf tiruan yang digunakan dengan langkah-langkah sebagai berikut [5] :

1. Menentukan variabel masukan.
2. Jumlah masukan ditentukan berdasarkan data jumlah penggunaan *bandwidth* di Universitas Pattimura.
3. Menentukan jumlah *neuron* pada *hidden layer* dan *context layer*.
4. Banyaknya jumlah *neuron* yang digunakan pada *hidden layer* dan *context layer* akan menentukan kinerja sistem ketika memprediksi penggunaan *bandwidth*. Hal ini disebabkan karena semakin banyak *neuron* dalam *hidden layer* dan *context layer* maka akan membuat *neuron* tersebut memiliki beban yang semakin sedikit dalam menampung data-data yang harus dipelajari. *Hidden layer* dan *context layer* ditentukan secara *trial* dan *error* sampai didapatkan hasil yang terbaik.
5. Menentukan parameter pembelajaran, yaitu maksimum *epoch*, toleransi *error* dan *learning rate*.
6. Menentukan fungsi bobot yang akan digunakan berdasarkan fungsi aktivasi. Penentuan nilai bobot minimum dan maksimum ditentukan atas dasar $0 < n < 1$ yang merupakan jangkauan dari fungsi. Fungsi ini digunakan untuk jaringan syaraf tiruan yang dilatih dengan menggunakan metode *Elman Recurrent*. Fungsi *sigmoid biner* mempunyai nilai antara 0 sampai 1. Fungsi ini sering digunakan jaringan syaraf tiruan yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1.
7. Menentukan *output*.

Nilai *output* ditentukan berdasarkan hasil yang diinginkan yaitu nilai berkisar antara 0 sampai 1 yang merepresentasikan nilai prediksi *bandwidth*, yang selanjutnya dinormalisasikan supaya menjadi nilai prediksi aktual.

3. Hasil dan Pembahasan

3.1 Analisis Preprocessing Data

Preprocessing data bertujuan untuk memilih data yang digunakan sebagai data penelitian, namun sebelumnya data tersebut di normalisasi terlebih dahulu. Dimana hasil normalisasi data tersebut selanjutnya di representasikan kedalam *sliding windows* dengan *windows size* sebesar lima. Proses *sliding windows* mengakibatkan data penelitian di reduksi karena proses penentuan data awal menjadi berubah. Total data

penelitian menjadi 82 vektor data, pada Tabel 1 menunjukkan contoh penerapan *windows size* pada data penelitian.

Tabel 1. Data penelitian dengan *windows size* 5

$x1'$	$x2'$	$x3'$	$x4'$	$x5'$	y'
0.313	0.246267	0.291533	0.245333	0.207067	0.2968
0.246267	0.291533	0.245333	0.207067	0.2968	0.252533
0.291533	0.245333	0.207067	0.2968	0.252533	0.142733
0.245333	0.207067	0.2968	0.252533	0.142733	0.3354
0.207067	0.2968	0.252533	0.142733	0.3354	0.2636
...

Data penelitian dengan *sliding windows* 5 selanjutnya dikelompokkan menjadi dua kelompok, yaitu data *training* sebanyak 80% dari keseluruhan data dan data *testing* sebanyak 20% dari data penelitian. Dari data keseluruhan yang berjumlah 82 vektor data maka didapatkan data yang menjadi data *training* set adalah sejumlah 67 data dan data yang menjadi data *testing* set sejumlah 15 data.

3.2 Training dengan nilai *window size*

Percobaan pertama untuk mendapatkan nilai *windows size* yang tepat yang menjadi parameter pengujian agar didapatkan hasil yang memiliki tingkat akurasi yang tinggi. Terdapat beberapa percobaan yang dilakukan, dan hasil yang dijadikan sebagai acuan awal adalah jumlah *epoch* maksimal yaitu 100.000 dan 1.000.000. Pelatihan akan berhenti jika telah mendapatkan $MSE < target\ error$ atau telah mencapai maksimum *epoch* yang ditentukan. Parameter *training* untuk pengujian nilai *windows size* dapat dilihat pada Tabel 2.

Tabel 2. Parameter *training* nilai *windows size* dengan *epoch* 100.000

Parameter	Nilai	Keterangan
Target error (MSE)	0,001	Target proses berhenti
Learning rate	0,1	Kecepatan pembelajaran sistem
Jumlah Hidden	3	Jumlah lapisan tersembunyi
Maksimum epoch	100.000	Jumlah iterasi
Transfer Function	Sigmoid Binner	
Bobot	Random (0,1)	

Hasil training dari percobaan ini adalah nilai dari *windows size* yang digunakan untuk mencari nilai MSE terkecil dengan *epoch* proses yang telah ditentukan. Nilai MSE terkecil yang didapatkan adalah 0.003277 dengan jumlah epoch sebanyak 100.000 *epoch*, hasil ini didapatkan pada nilai *input/windows size* delapan.

3.3 Training dengan jumlah neuron pada *hidden layer*

Jumlah *neuron* sangat mempengaruhi pada keakuratan hasil prediksi terhadap data yang digunakan dalam jaringan *Elman Recurrent Neural Network*. Jumlah *neuron* pada *hidden layer* yang cocok digunakan untuk prediksi *bandwidth* diperoleh melalui proses percobaan dengan melakukan beberapa kali *training*. Sebagai acuan awal *learning rate* yang digunakan adalah 0.01 dan 0.02. Target error adalah 0,001 dengan *epoch* maksimal sebanyak 1.000.000 dimana proses akan berhenti apabila nilai fungsi kinerja kurang dari atau sama dengan target error yang telah ditentukan atau telah mencapai nilai maksimum *epoch*.

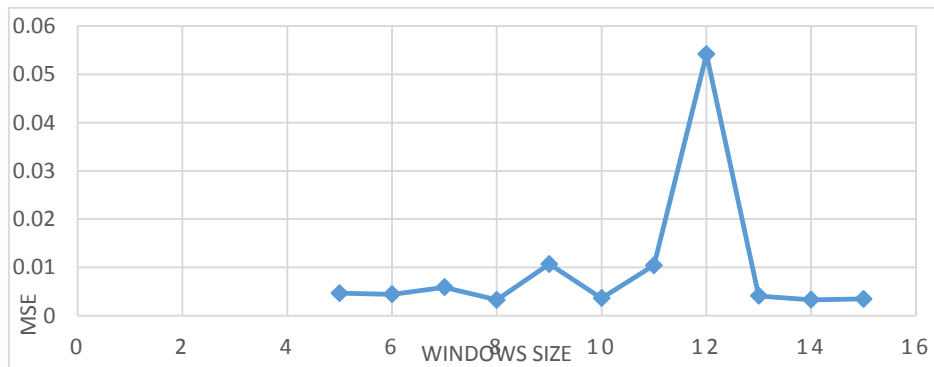
Tabel 3. Parameter pengujian jumlah neuron dengan target error 0.001

Parameter	Nilai	Keterangan
Target error (MSE)	0,001	Target proses berhenti
Learning rate	0,1	Kecepatan pembelajaran sistem
Jumlah Hidden	3,5,7,10,15,20, 25, 30	Jumlah lapisan tersembunyi
Maksimum epoch	100.000	Jumlah iterasi
Transfer Function	Sigmoid Binner	
Bobot	Random (0,1)	

Hasil percobaan yang didapat dengan menggunakan jumlah *neuron* yang berbeda-beda pada *hidden layer*, hasil yang didapat menunjukkan bahwa dengan jumlah *neuron* sebanyak tigabelas unit menghasilkan nilai *error* (MSE) terkecil dari seluruh percobaan yaitu sebesar 0.003725. Proses berikutnya adalah menguji hasil training dengan menggunakan jumlah *neuron* yang berbeda-beda pada tiap-tiap percobaan.

3.4 Analisis hasil training dengan *windows size*

Proses *training* yang dilakukan dengan menggunakan parameter *windows size* menunjukkan bahwa dengan jumlah *node input* 8 menghasilkan nilai MSE terkecil yaitu 0.003277. Berdasarkan Tabel 3 dibuatlah grafik perbandingan antara jumlah *neuron input* terhadap nilai MSE seperti pada Gambar 2.

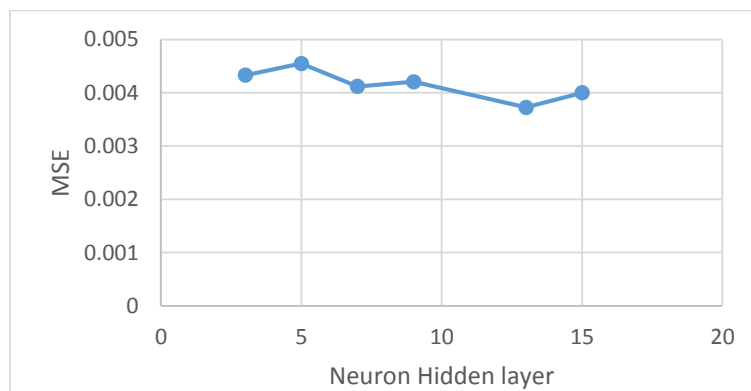


Gambar 2. Grafik perbandingan *windows size* dan nilai MSE (target 0.001)

Pada saat *training* dengan target 0.001 nilai *epoch* maksimum yang ditetapkan adalah 100.000 nilai MSE mengalami peningkatan pada saat jumlah *neuron input* 12 yaitu 0.054172.

3.5 Training dengan jumlah *hidden layer*

Pengujian dengan menggunakan parameter jumlah *hidden Layer* dengan parameter yang ditunjukkan pada Tabel 2. Berdasarkan Tabel 2 dibuatlah grafik perbandingan antara jumlah *windows size* terhadap nilai MSE seperti pada Gambar 3.



Gambar 3. Grafik perbandingan *neuron hidden layer* dan nilai MSE (target 0.001)

Pada saat *training* dengan *learning rate* 0.01 diperoleh konfigurasi terbaik dengan nilai MSE terkecil yaitu pada jumlah *neuron hidden layer* 13 yaitu sebesar 0.003725.

3.5 Analisis Hasil Testing

Pada penelitian ini data *testing* yang digunakan sebesar 20% dari total data yang digunakan, sehingga jumlah data yang digunakan sebanyak 15 vektor data. Data yang digunakan untuk *testing* merupakan data

baru yang tidak diikutsertakan dalam proses *training*. Tingkat akurasi hasil proses *testing* sangat dipengaruhi oleh bobot hasil *training*, yang menunjukkan kemampuan jaringan dalam mengenali pola-pola yang dilatih. Berdasarkan hasil percobaan yang dilakukan, dengan target toleransi *error* sebesar 0.001 dengan nilai masing-masing parameter sama dengan nilai yang terdapat pada proses *training* untuk jumlah *neuron hidden layer*.

Hasil *testing* dengan jumlah data sebanyak 15 data menghasilkan nilai MSE yang berbeda seperti yang ditunjukkan pada Tabel 4. Berdasarkan hasil *testing* diperoleh nilai MSE terkecil yaitu pada jumlah *neuron hidden layer* 13 yaitu sebesar 0.002422.

Tabel 4. Hasil testing dengan target error 0.001

<i>Neuron Hidden</i>	MSE
3	0.002513
5	0.002510
7	0.002526
9	0.002625
13	0.002422
15	0.002813

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut, sistem yang dikembangkan mampu mengenali pola dan dapat melakukan prediksi dalam hal penggunaan *bandwidth* dengan menggunakan metode jaringan saraf tiruan *elman*. Hasil *training* dengan menggunakan *windows* zise 8 pada maksimum *epoch* 100.000 diperoleh nilai MSE terkecil sebesar 0.003277. Hasil *training* dengan menggunakan *windows* zise 11 pada maksimum *epoch* 1.000.000 diperoleh nilai MSE terkecil sebesar 0.002833. Kemudian hasil *training* untuk jumlah *neuron* pada *hidden layer* diperoleh nilai MSE terkecil yaitu pada jumlah *neuron* 13 sebesar 0.003725. Hasil *testing* dengan menggunakan parameter pada percobaan pada jumlah *neuron hidden layer* 13 diperoleh nilai MSE terkecil yaitu sebesar 0.002422.

Daftar Pustaka

- [1] R. R. T. d. I. Muarifah, "Implementasi Mikrotik Sebagai Manajemen Bandwidth .," in *ISTAKPRIND*, Yogyakarta, 2012.
- [2] T. L. H. F. Pasila, F., "Elman Neural Network Application With Accelerated Lma Training For East Java-Bali Electrical Load Time Series Data Forecasting," 2009.
- [3] I. G. R. a. S. A. Mekongga, "The Prediction of Bandwidth On Need Computer Network Through Artificial Neural network Method of Backpropagation," vol. 02, p. 98–107., 2012.
- [4] M. S. S. A. R. Aziz, "Enhancement of Particle Swarm Optimization in Elman Recurrent Network with bounded Vmax Function," 2009.
- [5] Y. Noorviani, "Penerapan Elman Recurrent Neural network untuk Diagnosis Gangguan Autis Pada Anak," *Tesis, Universitas Gadjah Mada Yogyakarta*, 2011.
- [6] O. Abdalla, "A comparison of feed-forward back-propagation and radial basis artificial neural networks," *A Monte Carlo study. In Proceedings 2010 International Symposium on Information Technology - Engineering Technology*, p. 994–998, 2010.